ILLINOIS

UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

Final Report

Project 1: Trade @ U-I

CS 411

Dongni Yang NetID: dongniy2

Linfei Jing NetID: ljing2

Ruihao Zhang NetID: ruihaoz2

Ziyu Liu NetID: ziyuliu3

TA: Iris Du

Instructor: Professor Abdu

Aug.5th, 2020

Purpose and Usefulness

The project is capable of providing a platform for face-to-face second-hand goods trading around the campus town by letting users browse the details of commodities and contact sellers privately via email.

We found that during the graduation seasons or term begins seasons, there're lots of students willing to buy or sell second-hand items to save money, like sofas, mattresses, second-hand textbooks, or iclickers. When the semester starts, the demand for these items is often very urgent. If people choose to buy these things on eBay or Amazon, it usually will cost more than one week to wait for the items.

Therefore, second-hand trading between the same cities and even the same campus is very important. During the past, many students preferred to post the information on social media, such as Facebook and Twitter. However, these websites can't conveniently integrate these pieces of information; for example, these applications can't separate the goods by its prices and by its categories. People usually have to spend a long time browsing every post and find the information matches their requirements.

Our application provides an online platform to trade used goods based on the requirement of sellers and buyers. Unlike Ebay, we can give the users the opportunity to select locations for face-to-face exchanges in order to simplify the trading process and increase the reliability of buying the second-hand items.

We create the login and register system for the users to help them save the information. The users can create the products they want to sell and change the properties whenever they want. For the buyers, they can search the products they are interested in. When a product is sold out, the product can be deleted by the users. The users can also choose the city they prefer. Besides those, in our visions, the website, according to users' preferences, will provide some recommendations for users.

Data in Database

We collect the data from our friends, about 25 friends, 5 goods per person on average, since there is a lot of demand for selling out commodities during summer. Then by manually inputting the data, the database stores all the information which can be used in our project. The users will be asked to enter their contact information when they register, and the information about products, including the city of the transaction, the price, depreciation, and category of the products they want to sell.

ER Diagram and Schema

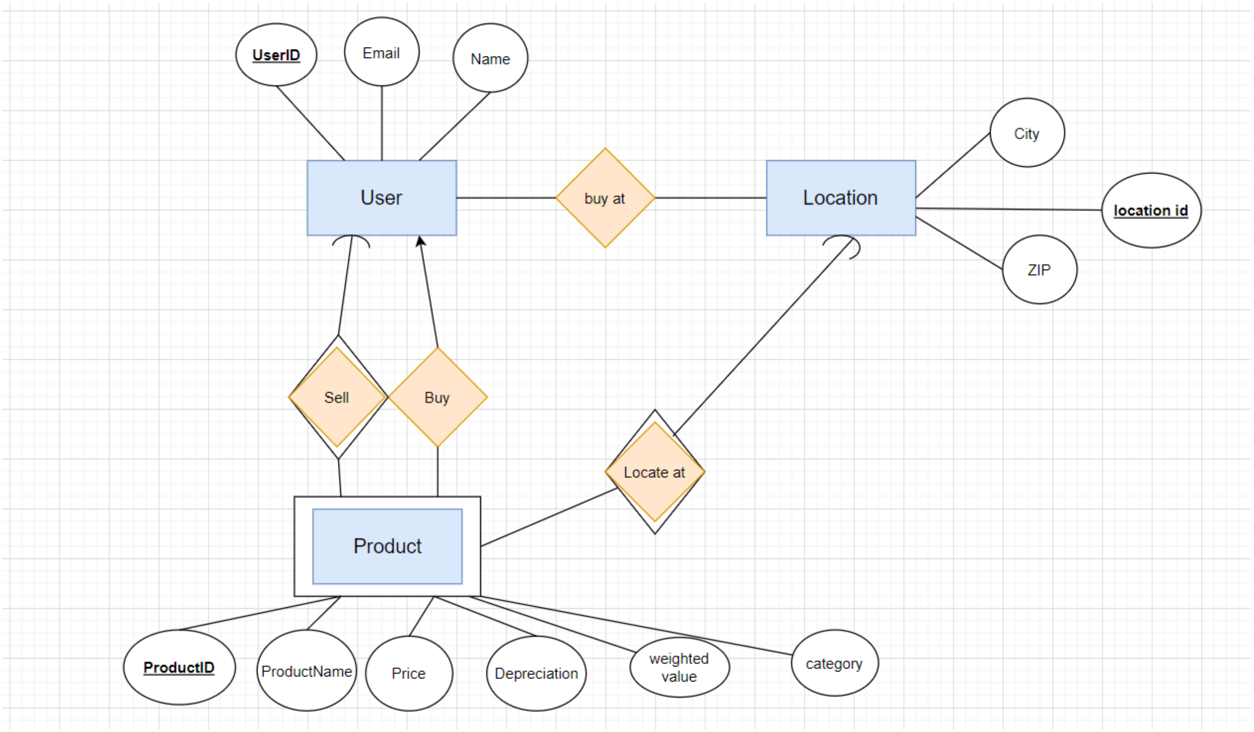


Figure1.ER Diagram

User(UserId->gender, UserName)

Name	Type
UserId	VARCHAR(225)
Email	VARCHAR(225)
UserName	VARCHAR(225)

Location(address->City, zip)

Name	Type
locationId	INT
City	VARCHAR(30)
zip	INT

Product(ProductId -> Price, ProductName, Depreciation)

Name	Type
ProductId	VARCHAR(225)
Price	REAL
ProductName	VARCHAR(225)
Depreciation	REAL
Weighted value	REAL
Category	VARCHAR(225)

Buy(UserId, ProductId)

Name	Type
UserId	VARCHAR(225)
ProductId	VARCHAR(225)

Functionality

The functionalities in our app are register, login, create products, delete products, update products, search products, and recommendation.

The register can help the users create their own account and save the account information to the local sql file.

The login can constrain the users to their own accounts and change the products in the account. And they are allowed to view others' products as well.

The create products can help users put more products they want to sell to the website. The users' operations will also be saved to the local sql file.

The delete products can help users delete the products they sold or they did not want to sell any more. The users' operations will also be shown in the local sql file.

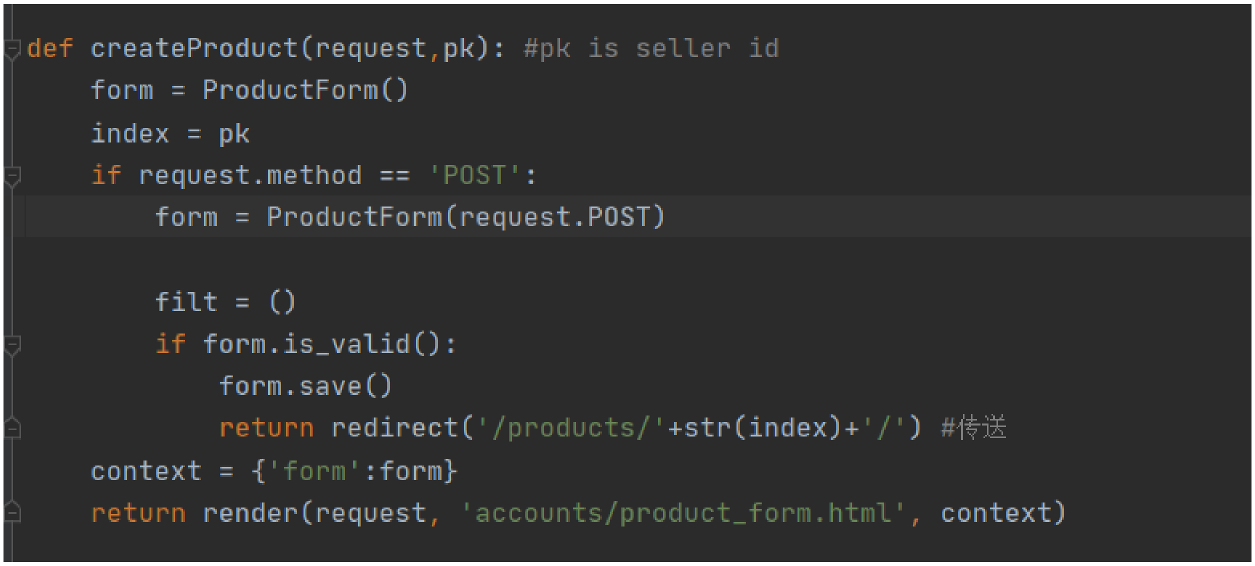
The update products can help users change products properties such as price. The users' operations will also be shown in the local sql file.

The search product can help the users find the products they via entering the product name in the searching bar.

The recommendation will implement search inside B+ tree and return the desirable commodities to the users via the preferences the users chose when they registered.

One Basic Function Explained

One basic function we implemented is CreateProduct, where the user is directed to an add-product webpage and asked to enter the product information. If the information is not completed, the webpage will stay at the current page; once the information is entered and completed, the method "POST" will be detected and the product will be created in the database and shown in the products webpage.



```
def createProduct(request,pk): #pk is seller id
    form = ProductForm()
    index = pk
    if request.method == 'POST':
        form = ProductForm(request.POST)

        filt = ()
        if form.is_valid():
            form.save()
            return redirect('/products/'+str(index)+'/') #传送
    context = {'form':form}
    return render(request, 'accounts/product_form.html', context)
```

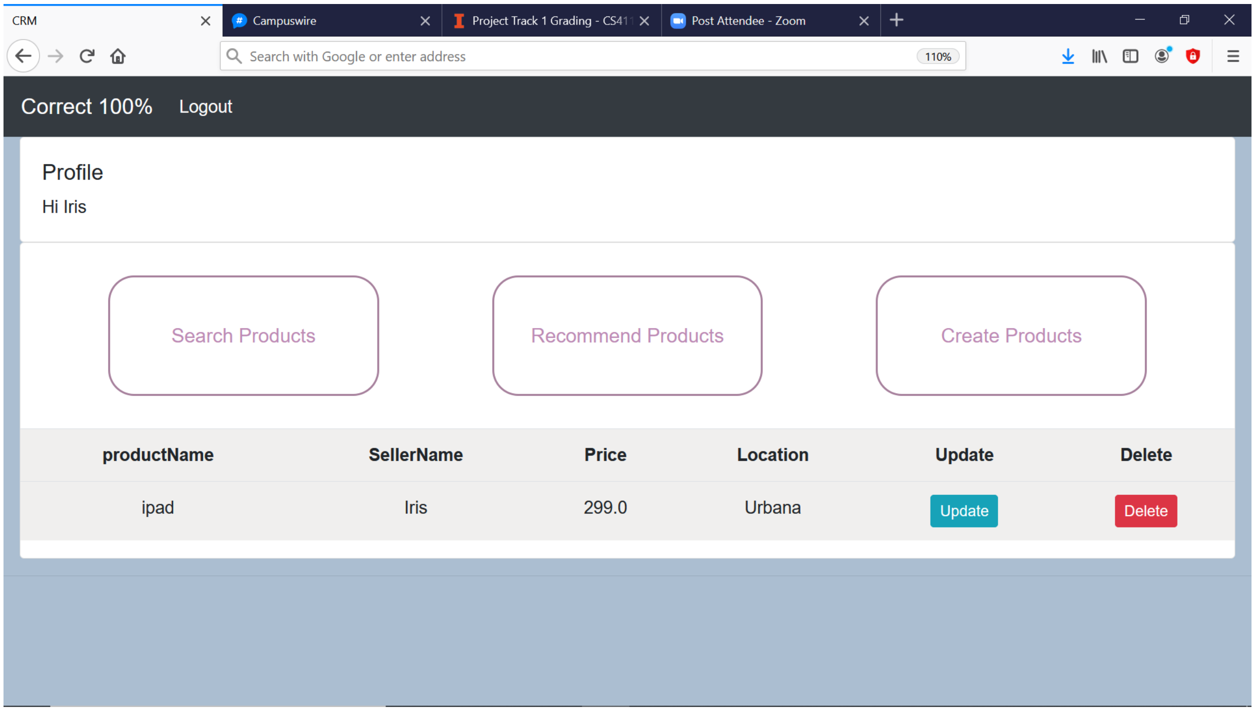
SQL Code

Insert Into accounts_product Values(“location”, “category”, “productName”, “price”, “depreciation”)

Dataflow

- Create New Product:

Before creating a new product, there’s one existing product, and after entering the data of the new product, the new product appears on the page under the associating account. On the database, the new product is already inserted in the product table.

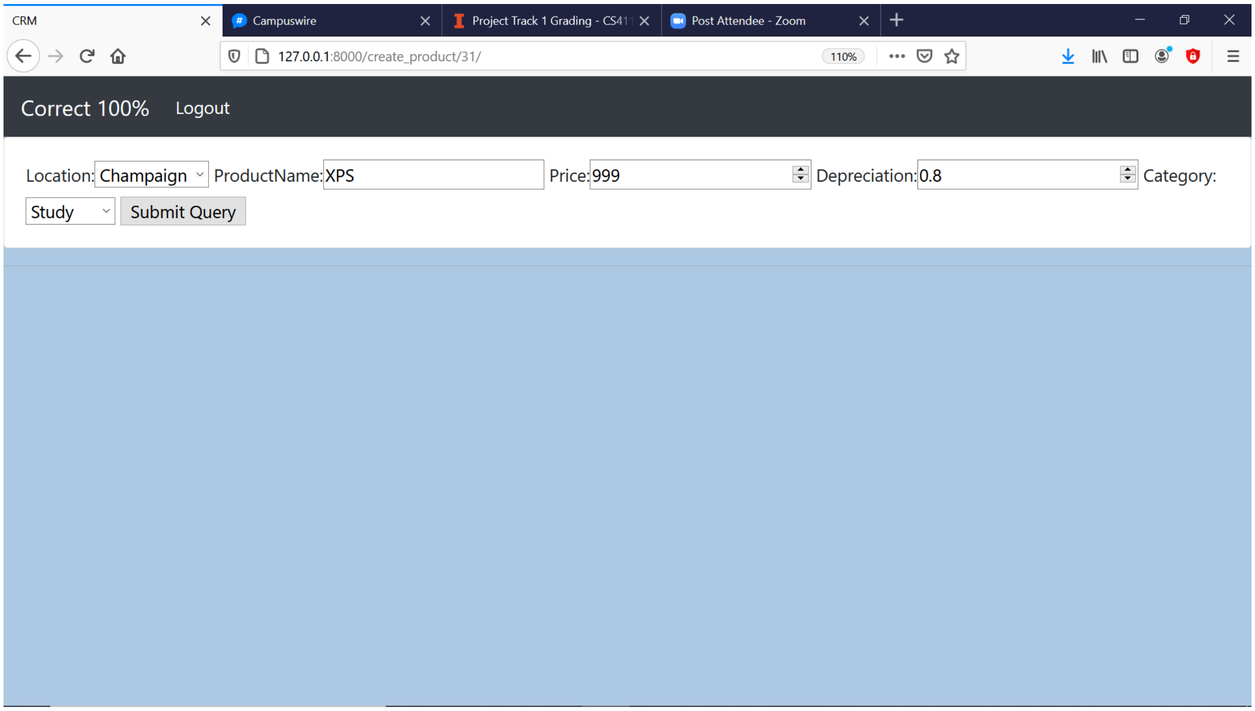


Profile

Hi Iris

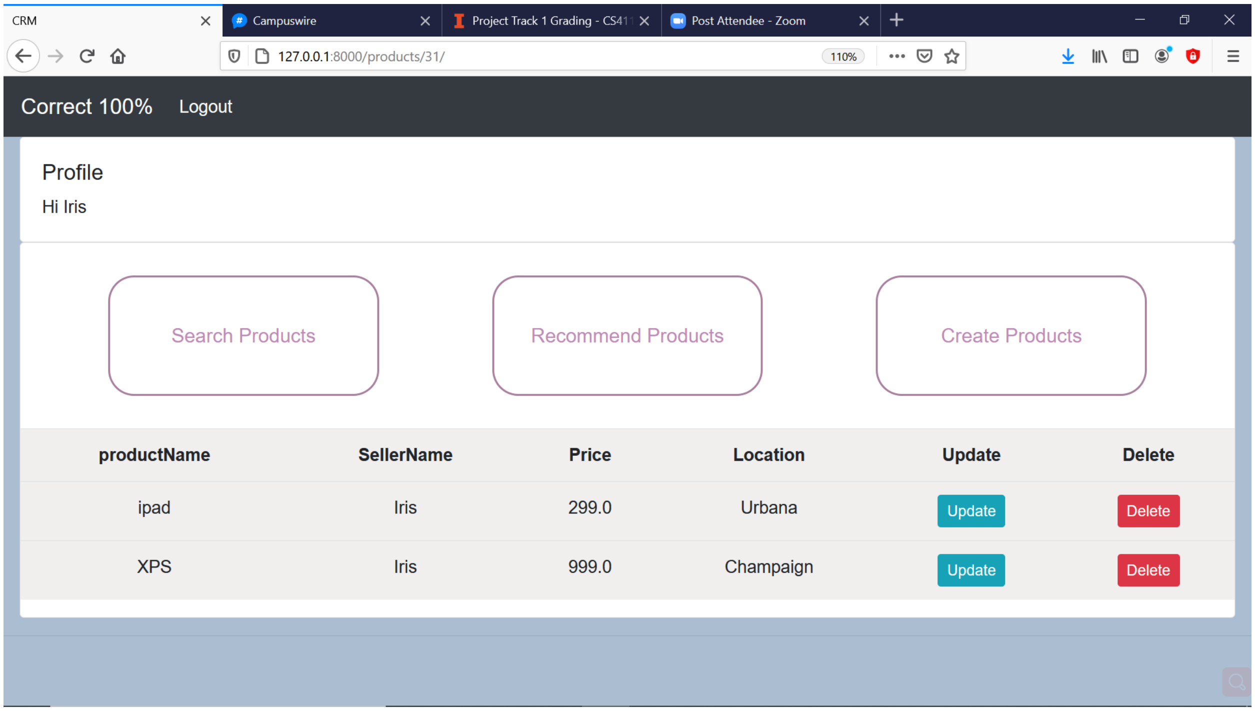
productName	SellerName	Price	Location	Update	Delete
ipad	Iris	299.0	Urbana	Update	Delete

Figure.Before Creating New Product


CRM
Campuswire
Project Track 1 Grading - CS41
Post Attendee - Zoom
127.0.0.1:8000/create_product/31/
110%
Correct 100%
Logout

Location: Champaign ProductName: XPS Price: 999 Depreciation: 0.8 Category: Study

Figure. Entering New Product



Profile

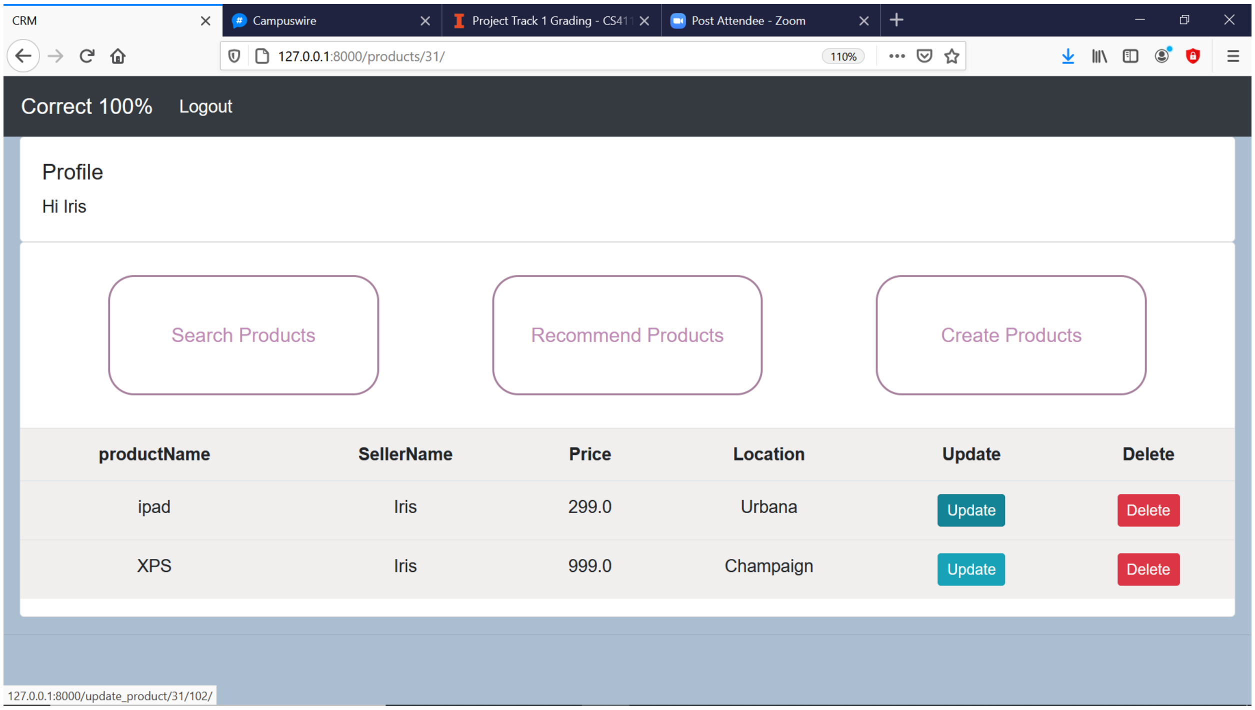
Hi Iris

productName	SellerName	Price	Location	Update	Delete
ipad	Iris	299.0	Urbana	Update	Delete
XPS	Iris	999.0	Champaign	Update	Delete

Figure. After creating New Product

- Update existed product:

Before changing a product, there's an existed product called ipad, and after changing the product's price to 199, the product's price on the page under the associating account is changed. On the database, the new product is already changed.

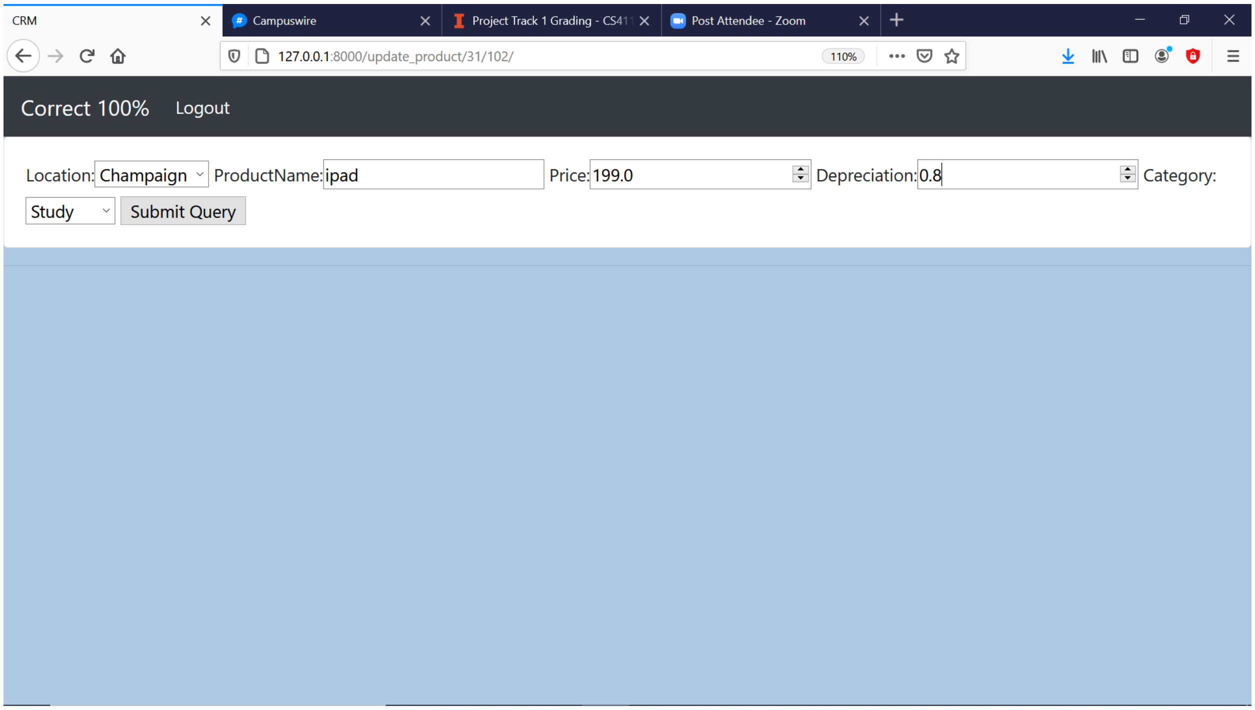


Profile

Hi Iris

productName	SellerName	Price	Location	Update	Delete
ipad	Iris	299.0	Urbana	Update	Delete
XPS	Iris	999.0	Champaign	Update	Delete

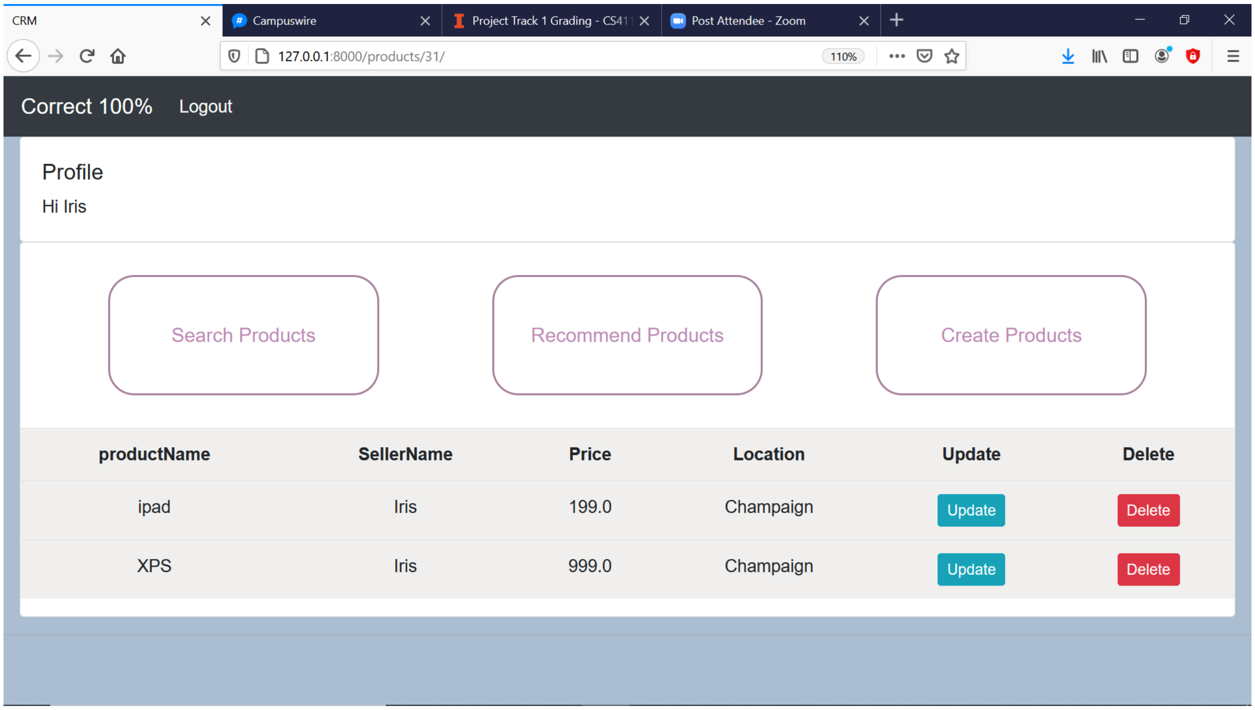
Figure.Before Updating Product



CRM
Campuswire
Project Track 1 Grading - CS41
Post Attendee - Zoom
127.0.0.1:8000/update_product/31/102/
110%
Correct 100%
Logout

Location: Champaign ProductName: ipad Price: 199.0 Depreciation: 0.8 Category: Study Submit Query

Figure: Updating Product



Profile

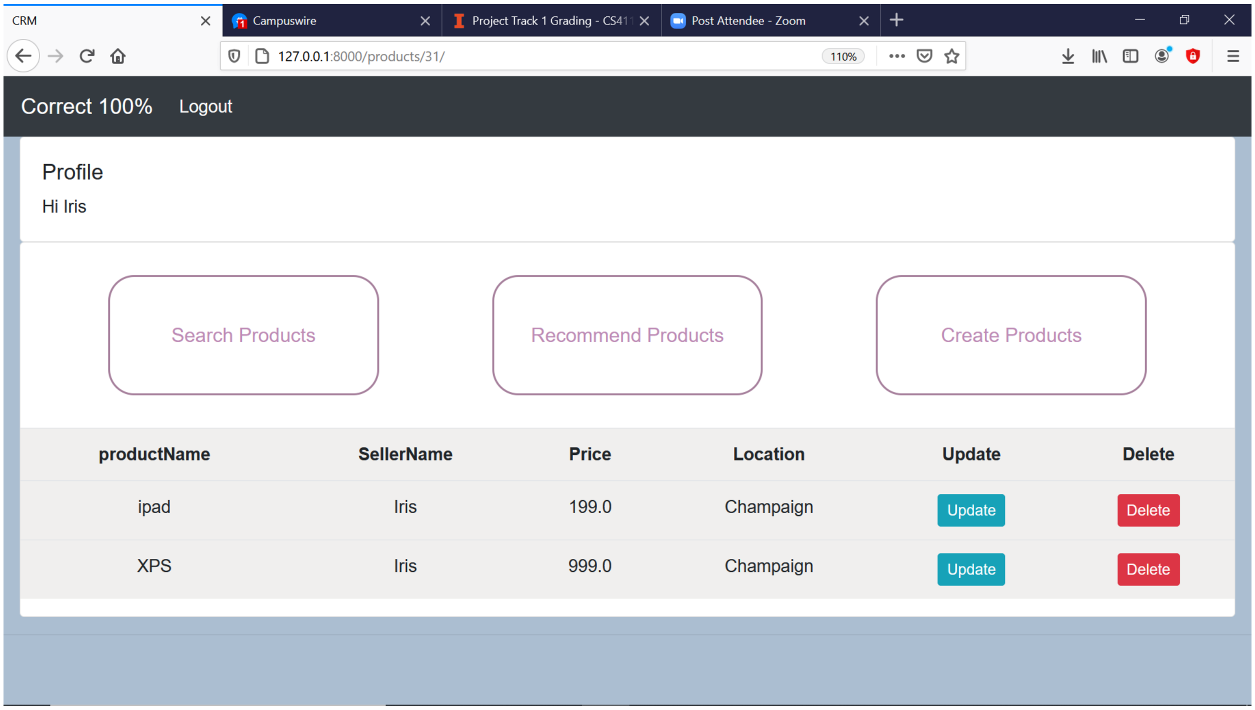
Hi Iris

productName	SellerName	Price	Location	Update	Delete
ipad	Iris	199.0	Champaign	Update	Delete
XPS	Iris	999.0	Champaign	Update	Delete

Figure: After updating the product

- Delete existed product:

Before changing a product, there’s an existed product called ipad, and after deleting the product, the product on the page under the associating account is deleted. On the database, the new product is already deleted.

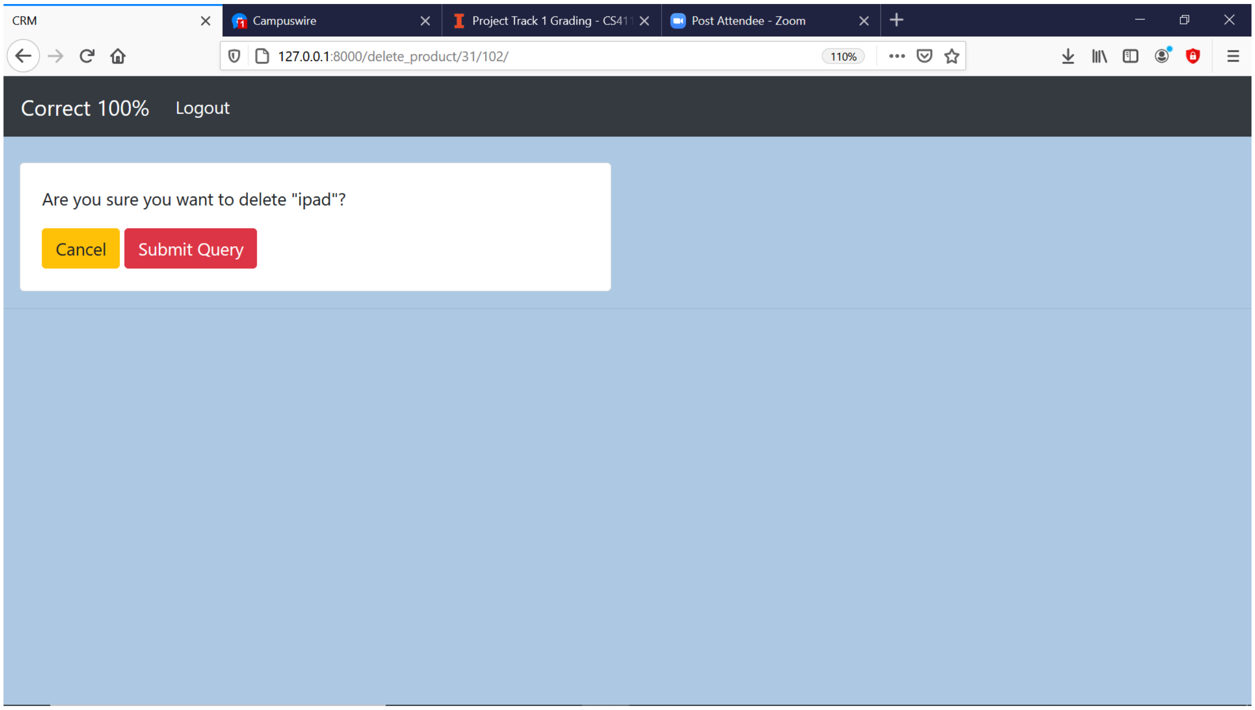


Profile

Hi Iris

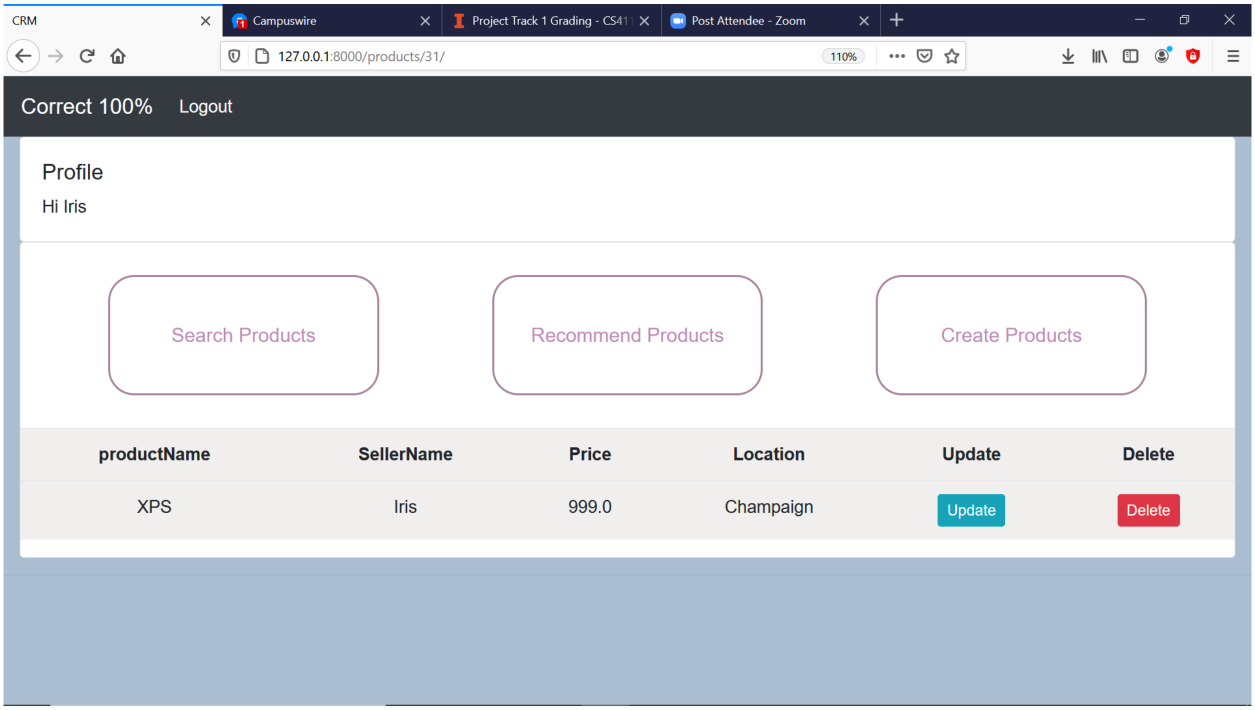
productName	SellerName	Price	Location	Update	Delete
ipad	Iris	199.0	Champaign	Update	Delete
XPS	Iris	999.0	Champaign	Update	Delete

Figure: Before deleting the product



Are you sure you want to delete "ipad"?

Figure: Deleting the product



Profile

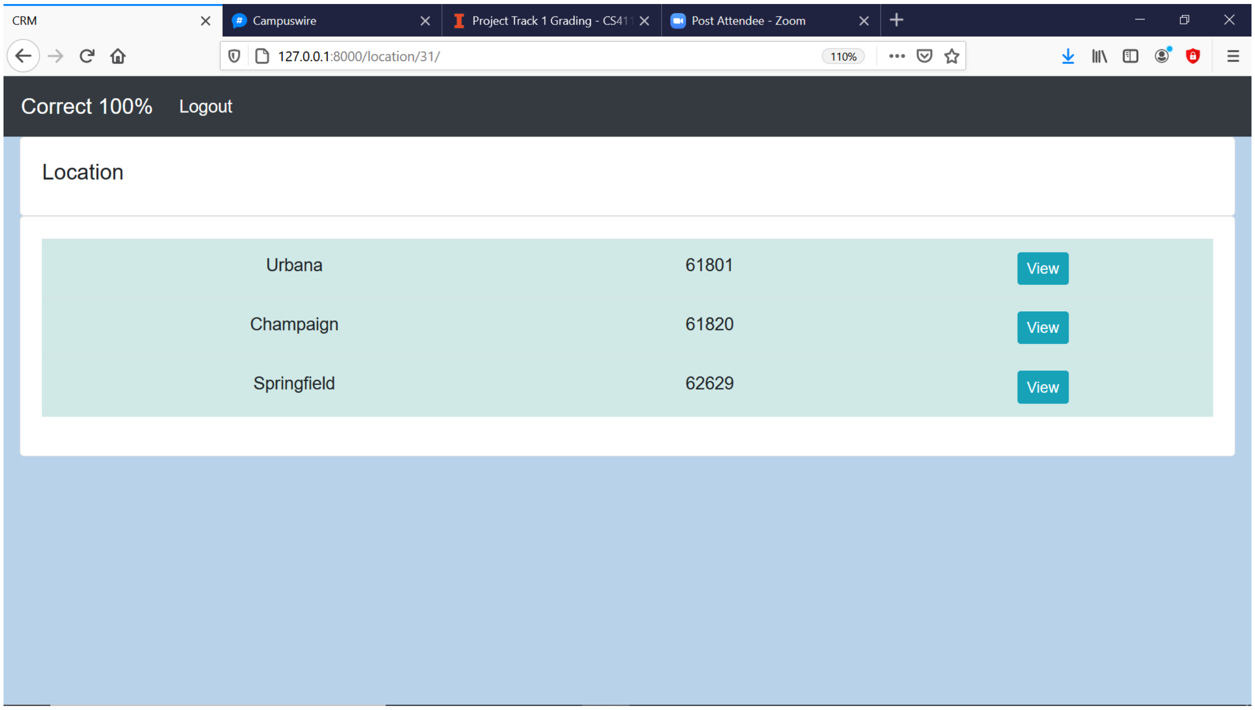
Hi Iris

productName	SellerName	Price	Location	Update	Delete
XPS	Iris	999.0	Champaign	Update	Delete

Figure: After deleting the product

- Search existed product:

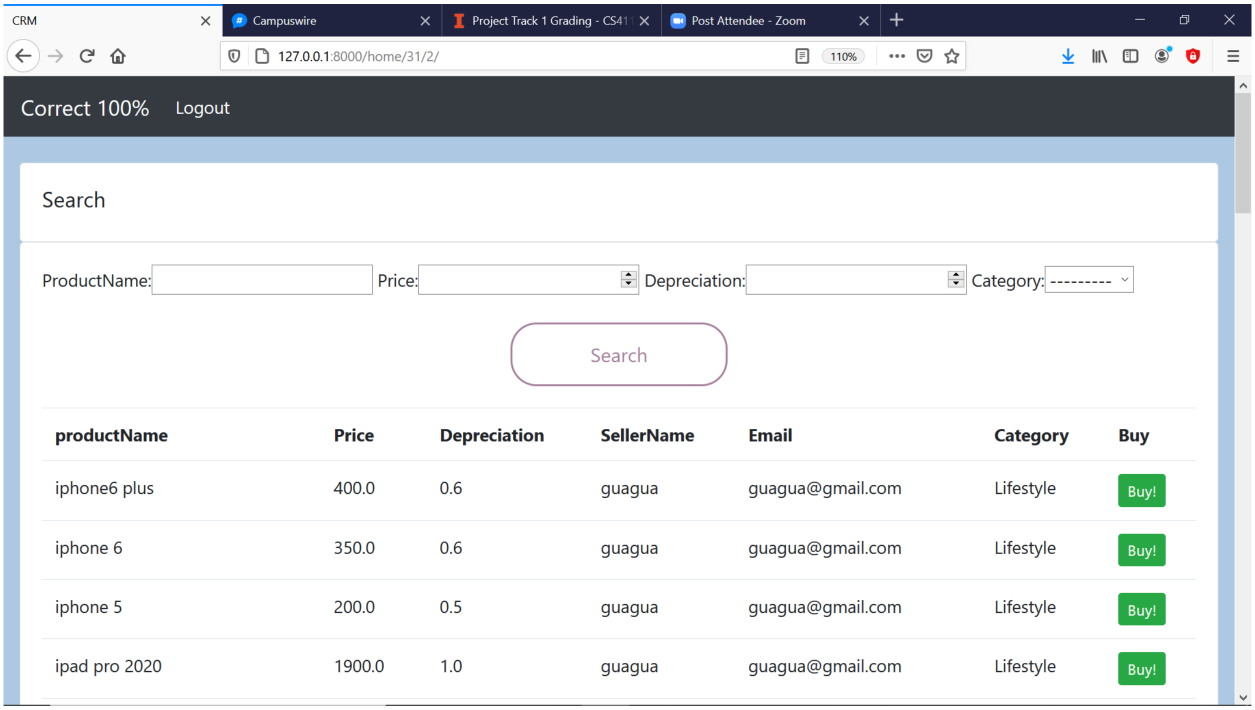
Here’s the search product function, the first step is to select the location. Since we want to design a face to face trading website. People can choose from Urbana, Champaign, and Springfield. All the products will be sorted by these three locations. After coming into the main page. People can search the products by its Name, Price, Depreciation, and Category. Whether a single search or combine search will be admitted.



Location

Urbana	61801	View
Champaign	61820	View
Springfield	62629	View

Figure: Choose the location before searching the product



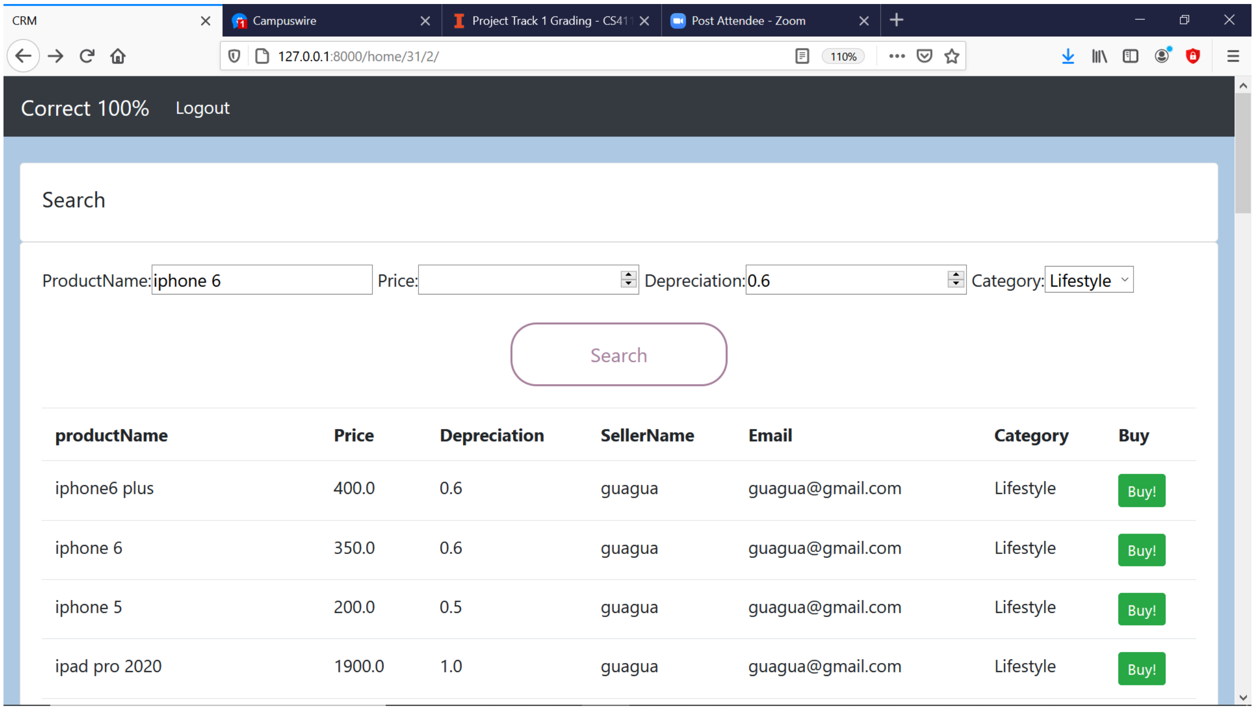
Search

ProductName: Price: Depreciation: Category: ---------

Search

productName	Price	Depreciation	SellerName	Email	Category	Buy
iphone6 plus	400.0	0.6	guagua	guagua@gmail.com	Lifestyle	Buy!
iphone 6	350.0	0.6	guagua	guagua@gmail.com	Lifestyle	Buy!
iphone 5	200.0	0.5	guagua	guagua@gmail.com	Lifestyle	Buy!
ipad pro 2020	1900.0	1.0	guagua	guagua@gmail.com	Lifestyle	Buy!

Figure: Before searching the product



Search

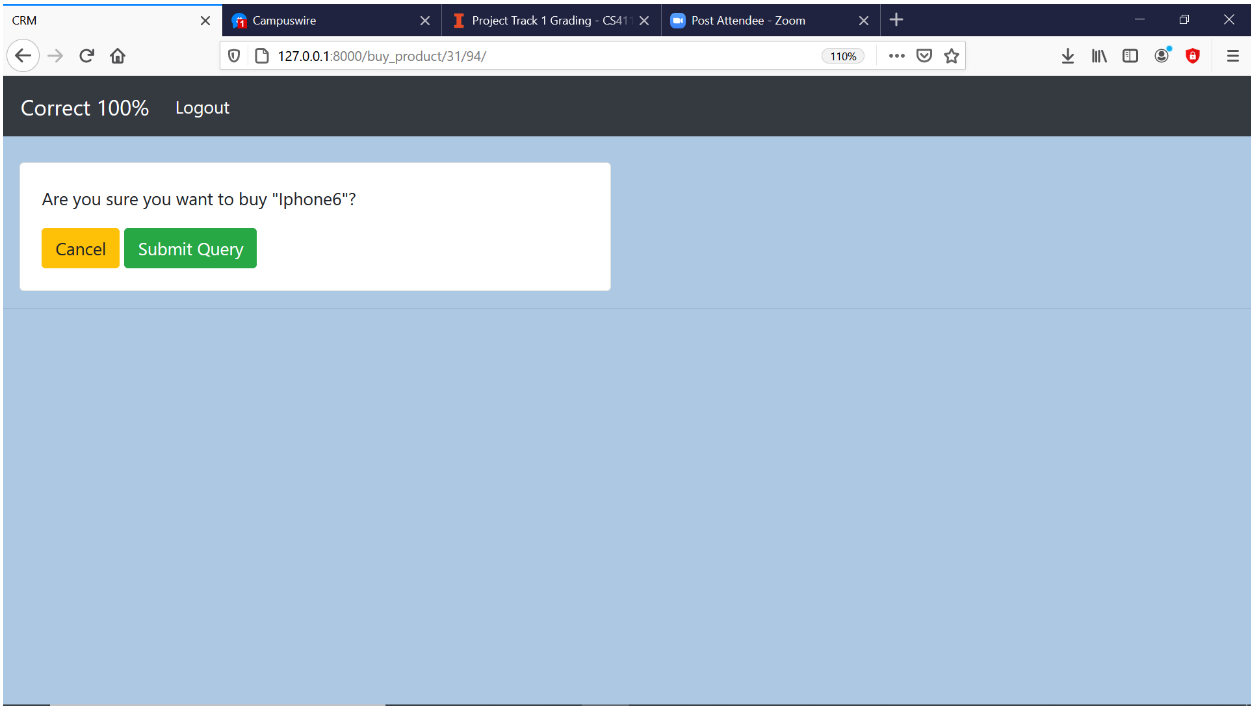
ProductName: iphone 6 Price: Depreciation: 0.6 Category: Lifestyle

Search

productName	Price	Depreciation	SellerName	Email	Category	Buy
iphone6 plus	400.0	0.6	guagua	guagua@gmail.com	Lifestyle	Buy!
iphone 6	350.0	0.6	guagua	guagua@gmail.com	Lifestyle	Buy!
iphone 5	200.0	0.5	guagua	guagua@gmail.com	Lifestyle	Buy!
ipad pro 2020	1900.0	1.0	guagua	guagua@gmail.com	Lifestyle	Buy!

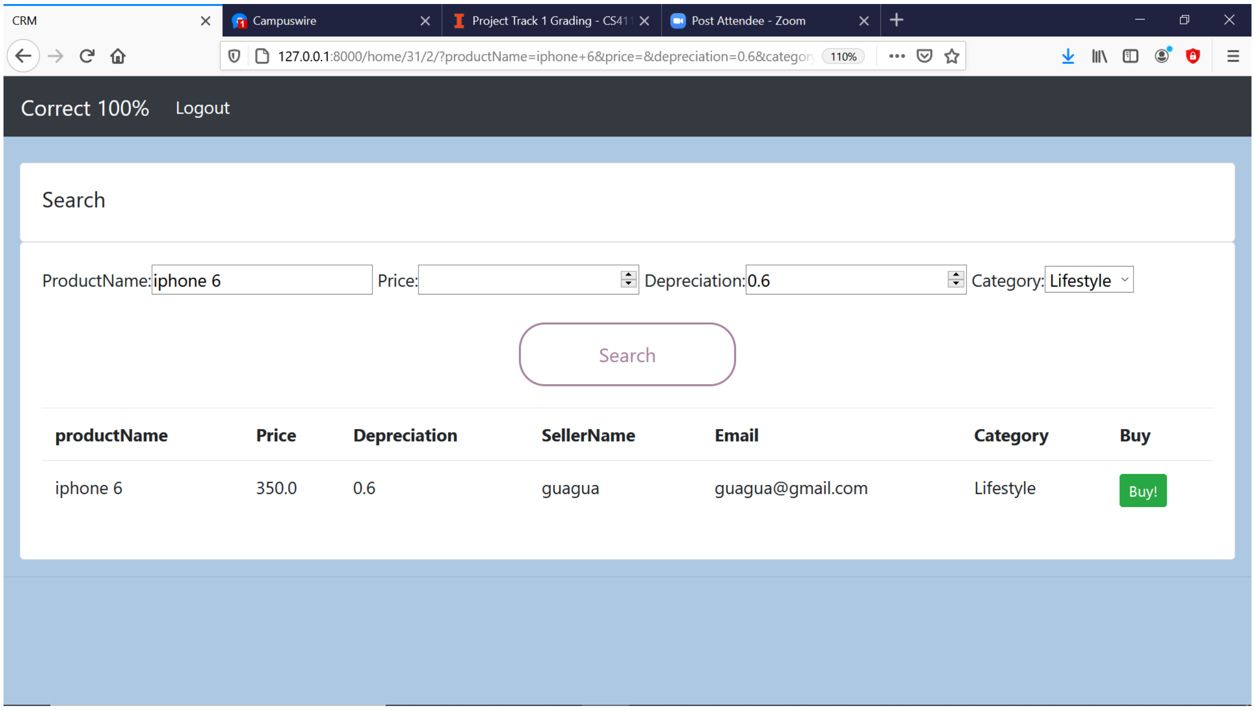
Figure: Searching the product

If you want to buy these products, just click the buy! Button and it will let people verify their selection. After buying these products. Follow the same search conditions again, people will find the products been deleted from the website.



Are you sure you want to buy "Iphone6"?

Figure: The ensuring page

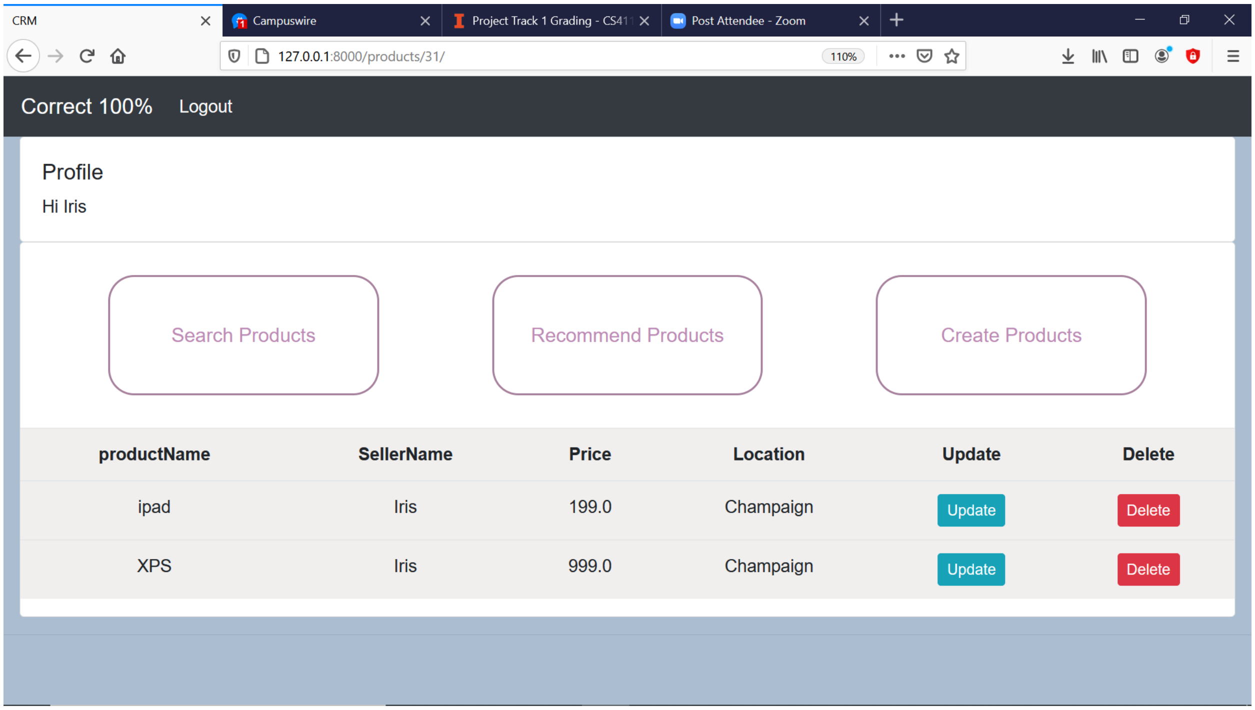

CRM
Campuswire
Project Track 1 Grading - CS41
Post Attendee - Zoom
127.0.0.1:8000/home/31/2/?productName=iphone+6&price=&depreciation=0.6&categor
110%
Correct 100%
Logout
Search
Search
Buy!

ProductName: iphone 6 Price: Depreciation: 0.6 Category: Lifestyle

productName	Price	Depreciation	SellerName	Email	Category	Buy
iphone 6	350.0	0.6	guagua	guagua@gmail.com	Lifestyle	Buy!

Figure: After the searching page

- Recommend user-preferred product



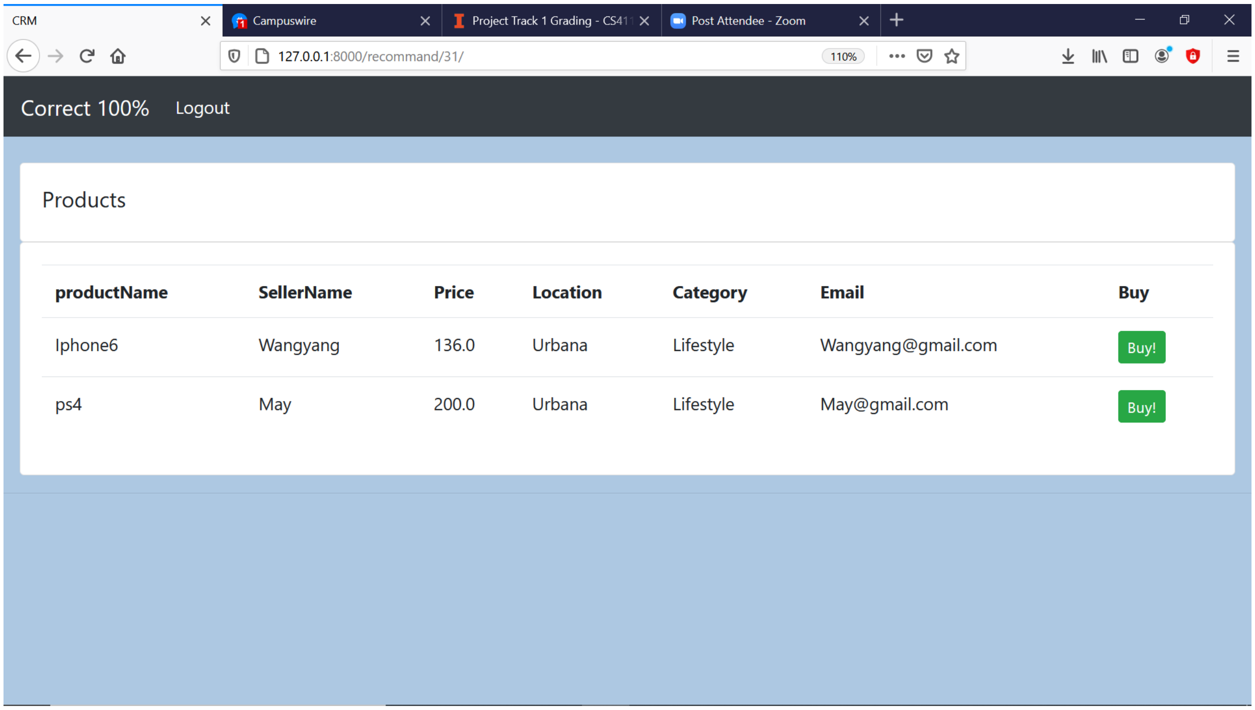
Profile

Hi Iris

productName	SellerName	Price	Location	Update	Delete
ipad	Iris	199.0	Champaign	Update	Delete
XPS	Iris	999.0	Champaign	Update	Delete

Figure: The main page

When the user clicks the "Recommendation Products", our application automatically calculates the "Product Value" and estimates the product value based on the user's preference. Then it would use the b+ tree to output the recommended product. Product Value is the leaf node of the b+ tree, and the website can find the corresponding tuple by the estimated number.

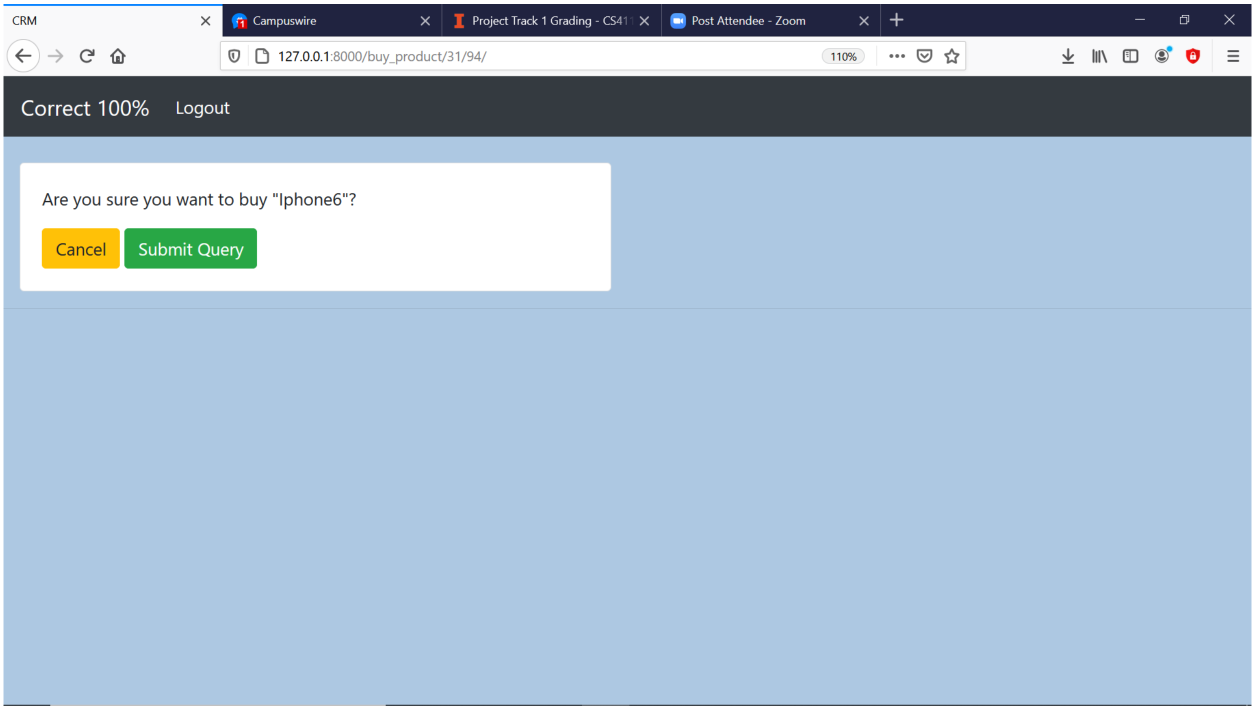


Products

productName	SellerName	Price	Location	Category	Email	Buy
Iphone6	Wangyang	136.0	Urbana	Lifestyle	Wangyang@gmail.com	Buy!
ps4	May	200.0	Urbana	Lifestyle	May@gmail.com	Buy!

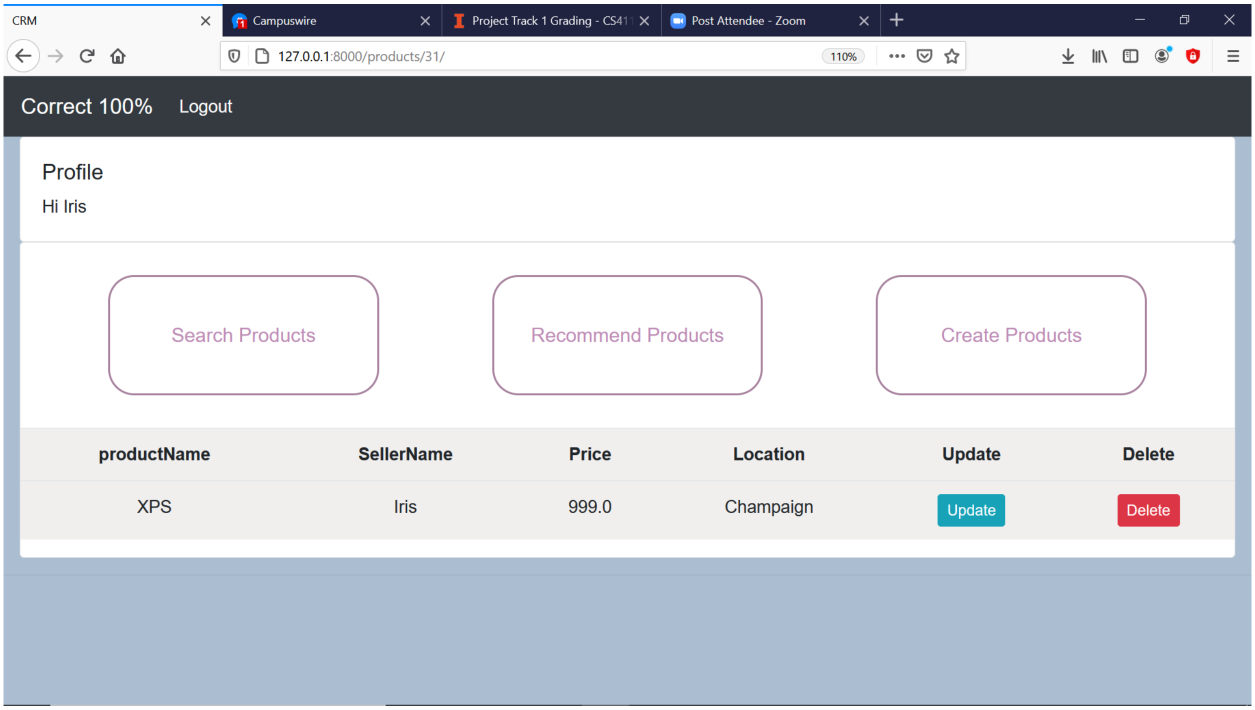
Figure: The recommendation page

The user can buy the product by clicking the green button on the right side. Then a new webpage should come out to ensure user's choice, and the webpage would send the data to the back-end and delete the products. Therefore, the product page will change.



Are you sure you want to buy "Iphone6"?

Figure: The ensuring page



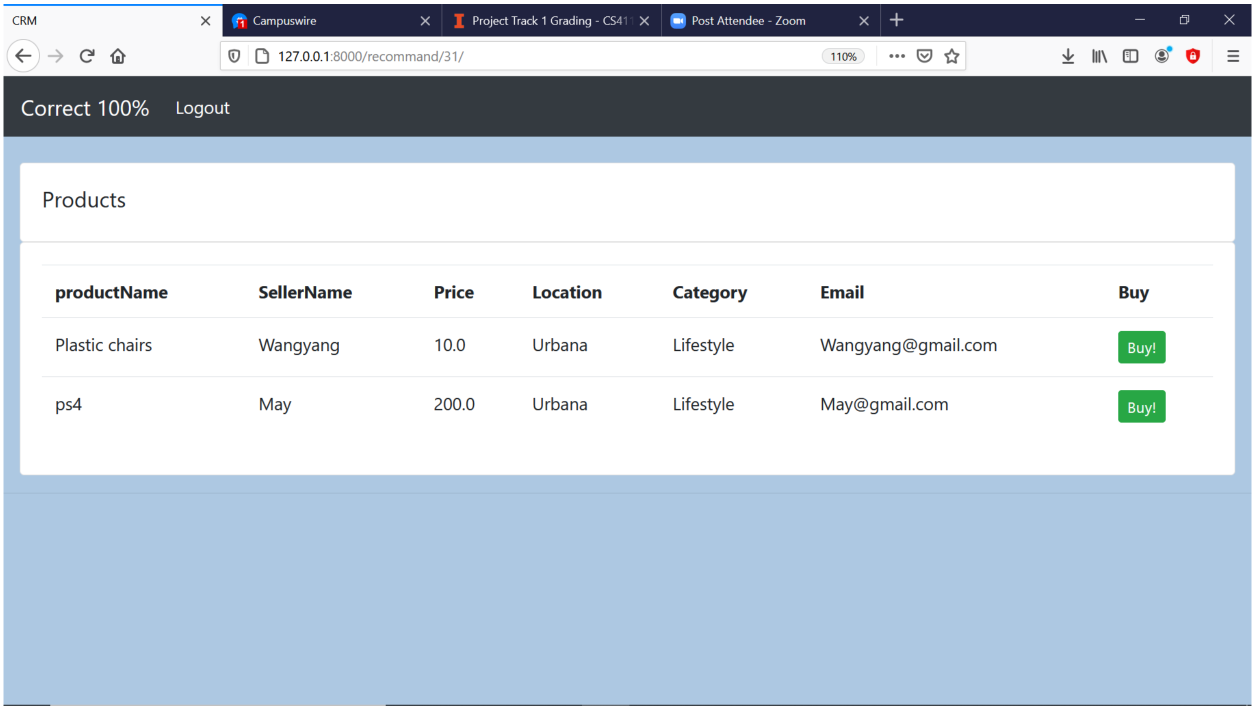
Profile

Hi Iris

productName	SellerName	Price	Location	Update	Delete
XPS	Iris	999.0	Champaign	Update	Delete

Figure: The main page after buying recommended product

Therefore, the product page will change. If users want to know the recommended products, they can click on the button again and get new recommendations.



Products

productName	SellerName	Price	Location	Category	Email	Buy
Plastic chairs	Wangyang	10.0	Urbana	Lifestyle	Wangyang@gmail.com	Buy!
ps4	May	200.0	Urbana	Lifestyle	May@gmail.com	Buy!

Figure: New recommendation page

Advanced Function

The advanced function we did is the recommendation function. We provided a unique weighted value for every products created, which is calculated via 2*location + 10*category + depreciation. When the same weighted value is found in the sql, we will add 0.1 until it is unique. And we will store the weighted values in a B+ tree. When the users register, they will make some choices. According to the users' choices, a weighted value will be created, which in the following is called customer value. The customer value will be searched from the B+ tree and return the values of blocks that the customer value should be inserted. And the products of the returned value will be recommended to the user. The reason why we thought it was the advanced function was because we implemented the B+ tree which could speed up the recommendation speed.

Challenges and Plan Changes

The problems we mainly faced were the lack of experiences. All four of us did not have the experiences about the website creating. So, we checked a lot of tutorial videos and learned from them such as how to connect the data in the website with the local sql database.

Something that we planned during the beginning of the project did not accomplish. Firstly, at the beginning, we hoped to do some visualization stuff. We hoped the buyers and sellers could know each other's location via the mark shown on the map. However, we found we needed third party data to accomplish this goal. Secondly, we wished to do the customer rating service. For both of the two goals, we found the workload was much heavier than expected and the time was so limited to finish those stuff for us.

Work Division

In our team, we are first splitted as two groups: two for front end and two for back end. After we finish the basic functions and basic structures of our project, we are then splitted into two groups based on our interest: two for advanced function, and two for further refining the webpage and improving the functionalities.

Video Link

https://youtu.be/IpTSK4iFEew